

How Far Can You Get By Combining Change Detection Algorithms?

Simone Bianco, Gianluigi Ciocca, and Raimondo Schettini

Abstract—In this paper we investigate how state-of-the-art change detection algorithms can be combined and used to create a more robust change algorithm leveraging their individual peculiarities. We exploited Genetic Programming (GP) to automatically select the best algorithms, combine them in different ways, and perform the most suitable post-processing operations on the outputs of the algorithms. In particular, algorithms' combination and post-processing operations are achieved with unary, binary and n -ary functions embedded into the GP framework. Using different experimental settings for combining existing algorithms we obtained different GP solutions that we termed IUTIS (*In Unity There Is Strength*). These solutions are then compared against state-of-the-art change detection algorithms on the video sequences and ground truth annotations of the ChandeDetection.net (CDNET 2014) challenge. Results demonstrate that using GP, our solutions are able to outperform all the considered single state-of-the-art change detection algorithms, as well as other combination strategies.

Index Terms—Change detection, algorithm combining and selection, genetic programming, CDNET.

I. INTRODUCTION

MANY computer vision applications require the detection of changes within video streams. For example, video surveillance applications need to track moving objects or identify abandoned ones to trigger event-related actions [1] by analyzing and monitoring the video content. Other applications that need video change detection are smart environments and video indexing and retrieval. For all these applications, a robust change detection algorithms with low false alarm rates is required as a pre-processing step. In the last decades, many algorithms have been proposed to solve the problem of video change detection [2]–[6].

The simplest strategy to detect foreground regions in video is to directly subtract the pixels in the current frame from those in a previous or reference one [7]. Clearly this approach, although very efficient is prone to many false positive errors due to video noise and global illumination changes. To limit these issues, temporal or adaptive filters can be applied to build the background model. For example median filter (e.g. [8]–[10]), Kalman filter [11]–[13], and a simplified version of Kalman filter called Wiener filter [14] have been applied. Pixel's values can also be analyzed in a given time slot using color histograms and considering the mode [15].

Another way to represent statistically the background is to consider the history over time of the values of the pixels. For example, the background can be modeled as a single Gaussian [16] or a Mixture of Gaussians [17]. The latter overcomes the limitation of the unimodal model that cannot handle dynamic background motion. The approaches using the Gaussian model can be also extended by incorporating the generalized Gaussian model [18], [19]. Also Bayesian approaches have been proposed to cope with backgrounds having large variations [20], [21]. For example, in [22], is proposed a Bayesian framework that incorporates spectral, spatial, and temporal features to characterize the background appearance in complex environments, while in [23] the model uses spatial statistics of the neighboring pixel values to robustly detect the foreground against

object's shadows. An hybrid moving object detection system that uses motion, change, and appearance information for more reliable detections is presented in [24]. The method called Flux Tensor with Split Gaussian models (FTSG) uses a split Gaussian method to separately model foreground and background.

The above statistical methods require the definition of the parameter's model. Non parametric methods directly rely on the observed data to statistically model the background [25]. Although these methods can deal with fast changes in the background, they are time consuming, and have an high memory requirement. Improvements have been proposed to overcome these problems e.g. [26]–[28].

Sample consensus is another non parametric strategy used to model background pixels. It has been recently used in ViBe [29] and SuBSENSE [30], [31]. Sample consensus determines if a given observation should be considered foreground or background based on its similarity to recently observed samples. The two works differ in the features used in describing and in that SuBSENSE introduces a feedback mechanism to continuously improving the pixel's modeling. In order to reduce the number of samples to be used to model the background, [32] exploits a small set of adaptive background templates. The templates are automatically discarded based on their estimated efficacy and new templates are added in their stead.

Subspace learning is another family of background modeling strategy. The frame image is considered as a whole and, by taking into account spatial information, they are more robust to illumination changes. For example, Oliver et al. [33] proposed an eigenbackground model. A set of images are used to build a vectorization representation of the scene within a given time frame. This representation is then decomposed via Principal Component Analysis to determine the background via the most descriptive eigenvectors. Using a subset of eigenvectors makes the background more robust to illumination changes. The need to efficiently update the background model within the video sequence have generated many variants such as Incremental [34], Incremental Non-Negative Matrix Factorization [35] and robust Matrix Factorization [36].

Other methods try to learn information about background or foreground by using some machine learning technique. These techniques conveniently incorporate domain knowledge from available samples. For example, in [37], the normalized optical flow and normalized frame differences are used with a probabilistic SVM to build a background block classifier. In [38], color, gradient and Haar-like features are integrated to handle spatio-temporal variations for each pixel within a Kernel Density Approximation framework, while background subtraction is performed using SVM. To overcome the problem of providing large sets of positive and negative examples to the learner, [39] proposes a 1-class SVM method that is able to update the classifier's parameters online.

Also, neural network-based solutions have received considerable attention. For example, the background segmentation approach proposed by [40] relies on a Probabilistic Neural Networks combining a neural network for background modeling and a Bayesian classifier for pixel's foreground/background detection. In [41] the SC_SOBS algorithm models the background with the weights of a neural network. A modified version of the algorithm is proposed in [42]

S. Bianco, G. Ciocca, and R. Schettini are with the Department of Informatic Systems and Communications, University of Milano-Bicocca, Milano, 20126 Italy, e-mail: {bianco, ciocca, schettini}@disco.unimib.it.

where the neural network is used to specifically detect moving object for PTZ cameras. A weightless neural network is proposed in [41]. The approach is called CwisarDH and uses a buffer of pixel values to store previous foreground values in order to make the algorithm more robust against intermittent objects.

All the above mentioned approaches, based their analysis on visual features computed on the video frames, either at pixel's level or at higher resolution level. A physics-based change detection approach is proposed in [43]. It uses image formation models to computationally estimate a consistent physics-based color descriptor of the spectral reflectance of foreground and background surfaces.

Regardless of the rationale of the approach, any background subtraction technique goals is to segment the scene into foreground and background components while trying to cope with the challenges that can be found in a real-world videos such as high variation in environments conditions, illumination changes, shadows, camera-induced distortions and so on. The output of a background subtraction algorithm is thus generally noisy, with isolated pixels, holes, and jagged boundaries. Post-processing of the foreground component, ranging from simple noise removal to complex object-level techniques, has been investigated to improve the algorithm's accuracy. Results indicate that significant improvement in performance are possible if a specific post-processing algorithm and the corresponding parameters are set appropriately [44]. To cope with the variability of real-world videos, algorithms are becoming increasingly complex and thus computationally expensive. Parallelization of background subtraction algorithms on GPU is a possible way to make them usable in real time applications [45]–[47].

Notwithstanding the improvements, change detection algorithms have been demonstrated to perform well on some types of videos but there is no single algorithm that is able to tackle all the challenges in a robust way. This can be clearly seen in the ChangeDetection.net (CDNET) 2014 competition [48], [49] (which follows the CDNET 2012 competition [50]) where change detection algorithms are evaluated on a common dataset composed of different types of videos sequences and classified according to their performance. As a consequence a combination of multiple change detection algorithms can be considered in order to achieve better overall results.

Several attempts to combine the outputs of different change detection algorithms have been investigated. For example the creators of the CDNET challenge have evaluated the performances of the top-3, top-5 and all the 28 reviewed using a simple majority vote fusion strategy [49]. The first two fusion schemes obtained the best results in the seven performances measure with respect to the top-ranked algorithm and even with respect to the fusion of all the 28 tested algorithms. Also in [51] a fusion scheme using majority vote is explored. In this case, the results of 22 algorithms have been fused as well as subsets of 3, 5 and 7 methods. Results show that the combination of different algorithms perform better with respect to single ones.

The problem of combining the change detection algorithms is similar to the problem of classifier combining or fusion. By considering the output masks generated by the algorithms as responses of pixel-based classifiers, the problem can be seen as selection and combination of, in our case, binary classifiers (foreground vs. background). Several works studied the problem of classifier combining or fusion (e.g. [52]–[57]).

Also in the context of image segmentation, the fusion of different algorithm's outputs is often exploited to obtain a more robust segmentation algorithms. For example, in [58], several fusion rules are investigated to improve segmentation accuracy. Specifically, the Median, Mean, Product, Minimum, and Maximum rules are considered, with the mean rule obtaining the overall best performances on the

segmentation datasets considered.

If the outputs of the classifier can be expressed as posteriori probabilities, a Bayesian methodology can be used to integrate the belief measure associated with each classifier to provide a combined final belief [59]. For example in [60], [61] the STAPLE algorithm uses an Expected Maximization strategy for estimating the “ground truth” segmentation from a group of expert segmentations in the context of medical imaging. STAPLE takes different segmentation and simultaneously estimates the final segmentation and the sensitivity and specificity parameters characterizing the performance of each expert. A similar approach is also used in [62] to estimate the final segmentation from atlas-based segmentations of three-dimensional confocal microscopy images of bee brains. In [63] the PRIF fusion scheme is presented. This scheme is based on a Markovian Bayesian fusion procedure, and the fusion is guided by the Probabilistic Rand Index [64]). This index measures the agreement of one segmentation result to multiple ground-truth segmentations, in a quantitative and perceptual way.

Recently [65] formalizes the problem of image segmentation fusion as a combinatorial optimization problem in terms of information theory. To reduce the computational complexity required with respect to previous optimization techniques a generative Bayesian image segmentation fusion model (BISF) is proposed.

In this paper we investigate how to create a new change detection algorithm by combining existing ones in a smart way. We aim at obtaining a new algorithm which is robust for different computer vision applications. To build our change detection algorithm from the results of existing ones, we rely on Genetic Programming [66]. As input we feed it the set of the binary foreground/background masks that correspond to the outputs of the change detection algorithms, and a set of unary, binary, and n -ary functions to perform both mask's combination (e.g. logical AND, logical OR) as well as mask's post-processing (e.g. filter operators). The solution tree obtained by Genetic Programming will give our change detection algorithm.

The advantage of using Genetic Programming is threefold. First, we are able to automatically select the algorithms that gives the best overall results over a set of predefined ones. Second, how to combine algorithms with which other ones to generate intermediate masks is automatically deduced. Third, which kind of post-processing of the original or intermediate masks to be applied in order to improve the results, is automatically built from the unary, binary and n -ary functions.

The organization of the paper is as follows. Section II illustrates how Genetic Programming is used to generate the combined change detection algorithm. In Section III-A we describe the experimental setup used in the evaluation of the proposed solution. Results are reported and discussed in Sections III-B. Finally Section IV concludes the paper.

II. GENETIC PROGRAMMING-BASED COMBINATION

Since there is no clear procedure to obtain a robust change detection algorithm by combining existing ones, a possible solution could be that of designing it by a trial-and-error process as done in [49] and [51] where different algorithms selections are tested. Instead, we propose an approach to automatically determine the best selection and combination of algorithms using Genetic Programming (GP) [66].

The major difference between GP and the other evolutionary algorithms is that GP is a domain-independent evolutionary method that genetically breeds a population of functions, or more generally, computer programs to solve a problem. In our case the solutions correspond to fusion strategies.

Algorithm 1: GP

```

1 begin
2   Generate a population  $P$  composed of an even number  $N$ 
   of individuals ;
3    $Generation \leftarrow 0$  ;
4   repeat
5     Calculate the fitness  $f$  of all the individuals in
     population  $P$  ;
6     Create a new empty population  $P'$  ;
7     repeat
8       Select two individuals  $C_i, C_j \in P$  using the chosen
       selection algorithm ;
9       Perform the crossover between  $C_i$  and  $C_j$  with
       probability  $p_c$ , and let  $\tilde{C}_i$  and  $\tilde{C}_j$  be the offspring. If
       crossover is not applied, let  $\tilde{C}_i = C_i$  and  $\tilde{C}_j = C_j$ ;
10      Mutate each character of  $\tilde{C}_i$  and  $\tilde{C}_j$  with certain
       probability  $p_m$ , and let  $\hat{C}_i$  and  $\hat{C}_j$  be the offspring;
11      Insert  $\hat{C}_i$  and  $\hat{C}_j$  into population  $P'$  ;
12    until until population  $P'$  is composed of exactly  $N$ 
       individuals;
13    Perform the copy  $P \leftarrow P'$  and delete  $P$  ;
14     $Generation \leftarrow Generation + 1$  ;
15  until a termination condition is satisfied;

```

The solutions can be represented as trees, lines of code, expressions in prefix or postfix notations, strings of variable length, etc. We use the representation first introduced in [66]: potential solutions are represented as LISP-like tree structures built using a set of terminal symbols \mathcal{T} and a set of nonterminal or functional symbols \mathcal{F} . The iterative process of a generic GP is given in Algorithm 1.

Before running GP, we need to set a number of parameters, which are as follows:

- The sets \mathcal{F} and \mathcal{T} of functional (or nonterminal) and terminal symbols that are used to build the potential solutions.
- The fitness function $f(\cdot)$.
- The population size N .
- The maximum size of the individuals, typically expressed as the maximum number of tree nodes or the maximum tree depth.
- The maximum number of generations.
- The algorithm used to initialize the population. A set of initialization algorithms can be found in [66]).
- The selection algorithm.
- The crossover rate.
- The mutation rate.
- Presence or absence of elitism (i.e. preserving unaltered the best solutions to the next iteration).

Given a set of n change detection algorithms $\mathcal{C} = \{C_i\}_{i=1}^n$, the solutions evolved by GP are built using the set of functional (or nonterminal) symbols \mathcal{F} and the set of terminal symbols $\mathcal{T} = \mathcal{C}$. The functional symbols corresponds to operations performed on the inputs. We explicitly incorporated into the GP framework the list of operations given in Table I along with their functional symbols. They operate in the spatial neighborhood of the image pixel, or combine (stack) the information at the same pixel location but across different change detection algorithms.

We define the fitness function used in GP taking inspiration from the CDNET website, where change detection algorithms are evaluated using different performance measures and ranked accordingly. Given a set of video sequences $\mathcal{V} = \{V_1, \dots, V_S\}$, a set of performance

TABLE I
THE SET OF FUNCTIONAL SYMBOLS USED IN GP AND THEIR
CORRESPONDING OPERATORS.

| Function | Inputs | Domain | Effect |
|----------|--------|---------|-----------------------------------------------------------------------|
| ERO | 1 | spatial | Morphological erosion with a 3×3 square structuring element |
| DIL | 1 | spatial | Morphological dilation with a 3×3 square structuring element |
| MF | 1 | spatial | Median filter with a 5×5 kernel |
| OR | 2 | stack | Logical <i>OR</i> operation |
| AND | 2 | stack | Logical <i>AND</i> operation |
| MV | >2 | stack | Majority vote |

measures $\mathcal{M} = \{m_1, \dots, m_M\}$ the fitness function of a candidate solution C_0 , $f(C_0)$ is defined as the average rank across video sequences and performance measures:

$$f(C_0) = \frac{1}{M} \sum_{j=1}^M \left(\text{rank}_{C_0} \left(C_0; \{m_j(C_k(\mathcal{V}))\}_{k=1}^n \right) + \sum_{i=1}^2 w_i P_i(C_0) \right) \quad (1)$$

where $\text{rank}_{C_0}(\cdot)$ computes the rank of the candidate solution C_0 with respect to the set of algorithms \mathcal{C} according to the measure m_j . $P_1(C_0)$ is defined as the signed distance between the candidate solution C_0 and the best algorithm in \mathcal{C} according to the measure m_j :

$$P_1(C_0) = \begin{cases} -m_j(C_0(\mathcal{V})) + \max_{C_k \in \mathcal{C}} m_j(C_k(\mathcal{V})) & \text{if the higher } m_j \text{ the better} \\ m_j(C_0(\mathcal{V})) - \min_{C_k \in \mathcal{C}} m_j(C_k(\mathcal{V})) & \text{if the lower } m_j \text{ the better} \end{cases} \quad (2)$$

and $P_2(C_0)$ is a penalty term corresponding to the number of different algorithms selected for the candidate solution C_0 :

$$P_2(C_0) = \frac{\# \text{ of algorithms selected in } C_0}{\# \text{ of algorithms in } \mathcal{C}} \quad (3)$$

The role of P_1 is to produce a fitness function $f(C_0) \in \mathbb{R}$, so that in case of candidate solutions having the same average rank, the one having better performance measures is considered a fitter individual in GP. The penalty term P_2 is used to force GP to select a small number of algorithms in \mathcal{C} to build the candidate solutions. The relative importance of P_1 and P_2 is independently regulated by the weights w_1 and w_2 respectively.

Our proposed combination strategy is able to simultaneously achieve three goals: algorithm selection, combination and processing. The set of functional symbols in Table I, act both as aggregation functions for the combination, as well as image processing functions (specifically the local functions such as the morphological ones and the median filter). Moreover, the nature of the GP algorithm coupled with the penalty factor P_2 , allow us to perform automatic algorithm selection in a seamless manner during the generation of the intermediate solutions. This is an advantage of our proposed strategy compared to other combination algorithms where the selection has to be performed in advance.

III. COMBINING STATE-OF-THE-ART ALGORITHMS

In this section we describe the experimental setup used in combining state-of-the art algorithms and the results. We based our

experiments on the CDNET 2014 challenge [48] that has received great attention in the evaluation of change detection algorithms. It provides a set of video sequences of various categories that can be used to test the algorithms on different environment conditions. Moreover, it provides an evaluation protocol that can be used to compare the performances of change detection algorithms against each other.

A. Experimental setup

As the set of algorithms to be combined we considered the top ranked ones that have been evaluated within the CDNET 2014 challenge as of July 2014. Table II shows the top 10 algorithms listed according to their average ranking across video categories. The outputs of the various algorithms are available on the website allowing us to perform our combination experiment on “certified” data. In our experiments, we selected a set of the top ranked algorithms, executed the GP algorithm using a training set of video sequences of the CDNET 2014 challenge, and then evaluated the obtained solution on the remaining video sequences. In particular, we created the GP combination algorithm using the shortest video sequences, one for each category. Specifically, the sequences used for training are (category/sequence): baseline/pedestrians, dynamicBackground/canoe, cameraJitter/badminton, intermittentObjectMotion/parking, shadow/peopleInShade, thermal/park, badWeather/wetSnow, lowFramerate/tramCrossroad_1fps, nightVideos/winterStreet, PTZ/zoomInZoomOut, and turbulence/turbulence3.

As for performance measures we computed them using the framework of the challenge that implements the following seven different measures: recall, specificity, false positive ratio (FPR), false negative ratio (FNR), percentage of wrong classifications (PWC), precision, and f-measure. A ranking of the tested algorithms is also computed starting from the partial ranks on these measures.

We evaluate our proposed combining strategy against three different fusion algorithms: a simple majority vote schema (MV), the STAPLE method [60], [61] (STAPLE), and a probabilistic rand index-based algorithm [63] (PRIF). Following the challenge’s rules, each algorithm uses a single set of parameters. The STAPLE algorithm estimates its parameter on-line (i.e. the sensitivities and specificities of each expert) while for the PRIF algorithm we used the default parameters. The parameters used to initialize the GP algorithm are reported in Table III.

B. Results

We applied GP to different sets of algorithms in \mathcal{C} using the top n algorithms in algorithms in Table II with $n \in \{3, 5, 7, 9\}$. The resulting algorithms, that we named IUTIS-3 and IUTIS-5 (quoting the Greek fabulists Aesop (620 BC-560 BC): “*In Unity There Is Strength*”), are shown in Figure 2 and Figure 3 respectively. In the same figures, for each solution tree, an example of the output at each node on a sample frame is also reported. From the solution trees it is possible to notice that IUTIS-3 automatically created MV-3 in its right branch. IUTIS-5, instead, automatically created as part of the solution MV-3 and MV-5 in its left and right branch respectively. It is worth nothing that thanks to the filtering and post-processing operations automatically selected by GP, the average rank of IUTIS-3 and IUTIS-5 are significantly lower than MV-3 and MV-5. In fact the average rank of IUTIS-3 is 3.41 points lower than that of MV-3, while IUTIS-5 is 4.42 points lower than that of MV-5.

The GP solutions with $n \leq 5$ (i.e. IUTIS-3 and IUTIS-5) used all the algorithms available in \mathcal{C} . The solutions obtained with $n > 5$ are not reported since in these cases the GP algorithm found a

solution identical to IUTIS-5. This is an evidence that GP is able to automatically select the best set of algorithms. This is one of the major difference between our method and the other fusion-based algorithms considered, which are not able to perform automatic algorithm selection, and thus use all the given algorithms. The effect is evident in Figure 1 where the average rank and F-Measure of the fusion-based algorithms on the CDNET 2014 challenge video sequences are reported: it is possible to see how the performance of PRIF and STAPLE decrease by increasing the number of algorithms available. MV-5 instead outperforms MV-3, but also in this case the performance decrease for $n = 7$ and 9. A different trend can be observed for IUTIS, which being able to perform automatic algorithm selection, is able to remove from the final solution those algorithms that could degrade the performance. The complete comparison of our proposed solutions to the single algorithms of Table II and fusion-based algorithms in the state-of-the-art in terms of ranking on the CDNET 2014 challenge video categories is reported in Table IV. It is possible to notice that all fusion-based algorithms with $n \leq 5$ outperform all the single algorithm, while this is true only for IUTIS and PRIF for higher values of n . In particular we can observe that there are three categories on which IUTIS-5 is not ranked first, i.e.: Turbulence, Dynamic Background and Intermittent Object Motion.

Outputs of some of the tested algorithms on sample frames in the CDNET 2014 dataset are shown in Figure 5 together with input images and ground truth masks. Detailed evaluation results of the IUTIS-3 and IUTIS-5 algorithms in terms of all the seven different measures for each category of the evaluation dataset are reported in Table V and VI respectively.

Since the proposed algorithm is stochastic by nature, we report in Figure 4 two variant solutions of found in different runs by our GP fusion scheme using the top 3 algorithms in Table II. These solutions trees are similar to IUTIS-3 and have the same overall results. With respect to IUTIS-3 they have introns, i.e. they possess non-functional branches: the sequence of erosions and dilations in the left branch of the top tree in Figure 4 is equivalent to a single erosion; in the bottom tree, the top OR operation is uninfuential on the final result.

Finally, Table VII reports the complete official ranking on the CDNET 2014 website at the moment of the submission (Nov 5 2015). As it can be seen, our solutions both outperform all the evaluated change detection algorithms including the most recent ones.

IV. CONCLUSION

In this paper we have presented an evolutionary approach, based on Genetic Programming, to combine change detection algorithms to create a more robust algorithm. The solutions provided by Genetic Programming allow us to automatically select the best subset of the input algorithms. This is one of the major difference between our method and the other fusion-based algorithms considered, which are not able to perform automatic algorithm selection, and thus use all the given algorithms. Moreover, we are able to automatically combine them in different ways, and perform post-processing on their outputs using unary, binary and n -ary operators embedded into the Genetic Programming framework.

We have shown that applying our approach on the best algorithms in the state-of-the-art, we are able to create the IUTIS-5 algorithm that ranks first by a large margin among a total of 29 change detection algorithms on the CDNET 2014.

As a future work we plan to investigate if the same approach, applied on very simple change detection algorithms, is able to create solutions with comparable results to more complex algorithm. Moreover, it would be interesting to use our framework to create new algorithms from scratch using atomic image processing operators as building blocks.

TABLE II
TOP 9 CHANGE DETECTION ALGORITHMS LISTED ON THE CDNET 2014 CHALLENGE WEBSITE (AS OF JULY 2014) RANKED BY THEIR AVERAGE RANKING ACROSS THE 11 VIDEO CATEGORIES.

| Rank | Method | Description | Reference |
|------|--------------|-----------------------------------------------------------------|-----------|
| 1 | FTSG | Flux Tensor with Split Gaussian models | [24] |
| 2 | SuBSENSE | Self-Balanced SENSitivity SEgmenter | [31] |
| 3 | CwisarDH | Change Detection with Weightless Neural Networks | [67] |
| 4 | Spectral-360 | Change Detection based on Spectral Reflectances | [43] |
| 5 | AMBER | Extension of the Adapting Multi-resolution Background Extractor | [32] |
| 6 | KNN | Adaptive Gaussian Mixture Model | [26] |
| 7 | SC_SOBS | Spatial Coherence Self-Organizing Background Subtraction | [41] |
| 8 | RMoG | Region-based Mixture of Gaussians | [68] |
| 9 | KDE | Change detection based on Kernel Density Estimation | [69] |

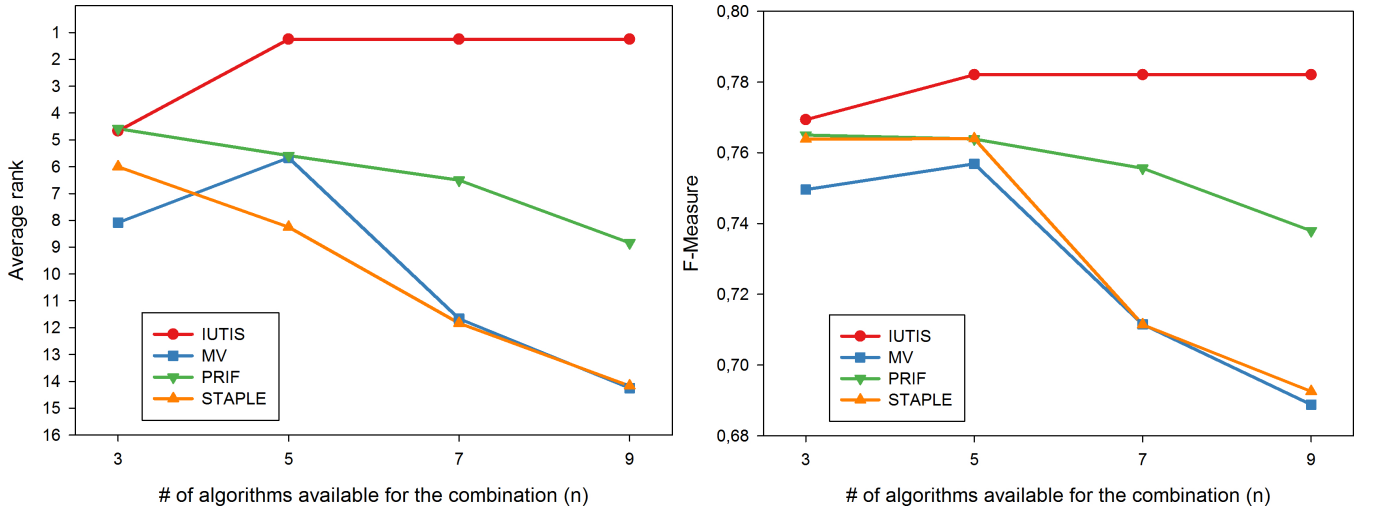


Fig. 1. Plots of the average rank (left) and F-measure (right) for the different fusion-based algorithms considered by varying the number of algorithms available in the combination, i.e. $n = 3, 5, 7, 9$

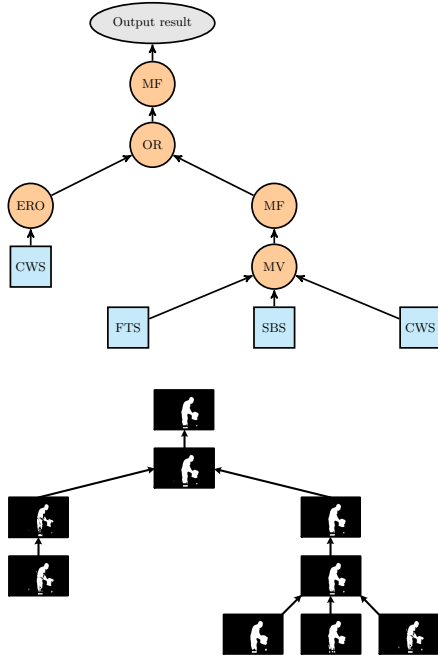


Fig. 2. IUTIS-3 solution tree and its example masks. SBS, FTS, and CWS refer to SuBSENSE, FTSG, and CwsarDH algorithm respectively.

TABLE III
SET OF PARAMETERS USED IN GP.

| Parameter name | Setting |
|---------------------------|--------------------------------------------------------------|
| Functional symbols | \mathcal{F} (see Sec. II) |
| Terminal symbols | \mathcal{T} (see Sec. II) |
| Fitness function | $f(\cdot)$ defined in Eq. 1 with $[w_1, w_2] = [0.01, 0.01]$ |
| Population size | 50 |
| Max tree depth | Dynamic |
| Max number of generations | 100 |
| Initialization algorithm | Ramped half-and-half [66] |
| Selection algorithm | Tournament, with tournament size 5 |
| Crossover rate | Adaptive [70] |
| Mutation rate | Adaptive [70] |
| Elitism | Yes |

REFERENCES

- [1] M. Sedky, M. Moniri, and C. Chibelushi, "Classification of smart video surveillance systems for commercial applications," in *Advanced Video and Signal Based Surveillance, 2005. AVSS 2005. IEEE Conference on*, 2005, pp. 638–643.
- [2] R. Radke, S. Andra, O. Al-Kofahi, and B. Roysam, "Image change detection algorithms: a systematic survey," *Image Processing, IEEE Transactions on*, vol. 14, no. 3, pp. 294–307, 2005.

TABLE IV

COMPARISON OF OUR PROPOSED SOLUTIONS TO SINGLE AND FUSION-BASED ALGORITHMS IN THE STATE-OF-THE-ART IN TERMS OF RANK IN EACH VIDEO CATEGORY AND AVERAGE RANK.

| Method ID | Avg rank | Overall | Bad Weath. | Low F.rate | Night Videos | PTZ | Turb. | Base. | Dynam. Backg. | Camera Jitter | Interm. Obj. M. | Shadow | Therm. |
|--------------|----------|---------|------------|------------|--------------|-----|-------|-------|---------------|---------------|-----------------|--------|--------|
| IUTIS-5 | 1.25 | 1 | 1 | 1 | 1 | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 1 |
| IUTIS-7 | 1.25 | 1 | 1 | 1 | 1 | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 1 |
| IUTIS-9 | 1.25 | 1 | 1 | 1 | 1 | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 1 |
| PRIF-3 | 4.58 | 2 | 3 | 2 | 3 | 3 | 5 | 3 | 9 | 7 | 6 | 6 | 6 |
| IUTIS-3 | 4.67 | 3 | 2 | 3 | 8 | 5 | 9 | 4 | 4 | 4 | 8 | 3 | 3 |
| PRIF-5 | 5.58 | 4 | 7 | 6 | 5 | 4 | 1 | 10 | 3 | 8 | 7 | 8 | 4 |
| MV-5 | 5.67 | 5 | 9 | 8 | 2 | 6 | 10 | 2 | 10 | 3 | 3 | 6 | 4 |
| STAPLE-3 | 6.00 | 6 | 4 | 5 | 4 | 8 | 6 | 5 | 5 | 8 | 12 | 2 | 7 |
| PRIF-7 | 6.50 | 8 | 10 | 4 | 9 | 2 | 3 | 9 | 1 | 6 | 16 | 5 | 5 |
| MV-3 | 8.08 | 7 | 11 | 11 | 6 | 13 | 8 | 6 | 12 | 5 | 4 | 5 | 9 |
| STAPLE-5 | 8.25 | 9 | 6 | 9 | 13 | 12 | 4 | 13 | 7 | 9 | 5 | 4 | 8 |
| PRIF-9 | 8.83 | 12 | 12 | 7 | 12 | 9 | 5 | 12 | 6 | 2 | 14 | 8 | 7 |
| FTSG | 9.67 | 11 | 8 | 14 | 10 | 11 | 12 | 17 | 8 | 16 | 1 | 6 | 2 |
| SuBSENSE | 10.67 | 10 | 5 | 10 | 7 | 10 | 7 | 14 | 15 | 15 | 17 | 7 | 11 |
| MV-7 | 11.67 | 13 | 14 | 8 | 14 | 14 | 13 | 7 | 15 | 12 | 9 | 11 | 10 |
| STAPLE-7 | 11.83 | 15 | 11 | 13 | 15 | 16 | 11 | 9 | 13 | 10 | 10 | 9 | 10 |
| STAPLE-9 | 14.17 | 16 | 15 | 15 | 17 | 17 | 15 | 8 | 16 | 13 | 15 | 10 | 13 |
| MV-9 | 14.25 | 17 | 13 | 12 | 17 | 15 | 14 | 11 | 17 | 14 | 13 | 16 | 12 |
| CwisarDH | 14.42 | 14 | 19 | 17 | 12 | 7 | 17 | 16 | 14 | 11 | 20 | 12 | 14 |
| Spectral-360 | 17.00 | 18 | 18 | 16 | 11 | 18 | 18 | 19 | 17 | 18 | 21 | 14 | 16 |
| AMBER | 17.33 | 19 | 16 | 22 | 21 | 23 | 16 | 22 | 11 | 13 | 11 | 15 | 19 |
| RMoG | 18.58 | 20 | 18 | 21 | 18 | 21 | 21 | 21 | 19 | 19 | 18 | 13 | 14 |
| SC_SOBS | 18.75 | 21 | 21 | 20 | 16 | 20 | 22 | 15 | 18 | 17 | 19 | 18 | 18 |
| KNN | 19.00 | 22 | 17 | 18 | 19 | 19 | 20 | 20 | 20 | 20 | 22 | 16 | 15 |
| KDE | 20.00 | 23 | 20 | 19 | 20 | 22 | 19 | 18 | 21 | 21 | 23 | 17 | 17 |

TABLE V

DETAILED EVALUATION RESULTS OF THE IUTIS-3 ALGORITHM FOR EACH CATEGORY OF THE EVALUATION DATASET.

| Scenarios | Recall | Specificity | FPR | FNR | PWC | Precision | FMeasure |
|----------------------------|--------|-------------|--------|--------|--------|-----------|----------|
| Overall | 0.7896 | 0.9944 | 0.0056 | 0.2104 | 1.1813 | 0.7951 | 0.7694 |
| Bad Weather | 0.7502 | 0.9993 | 0.0007 | 0.2498 | 0.5010 | 0.9280 | 0.8246 |
| Low Framerate | 0.8183 | 0.9964 | 0.0036 | 0.1817 | 0.8224 | 0.7813 | 0.7949 |
| Night Videos | 0.6243 | 0.9839 | 0.0161 | 0.3757 | 2.4354 | 0.4312 | 0.4814 |
| PTZ | 0.6508 | 0.9885 | 0.0115 | 0.3492 | 1.4869 | 0.3886 | 0.4230 |
| Turbulence | 0.7708 | 0.9998 | 0.0002 | 0.2292 | 0.1823 | 0.9368 | 0.8416 |
| Baseline | 0.9712 | 0.9981 | 0.0019 | 0.0288 | 0.3002 | 0.9393 | 0.9546 |
| Dynamic Background | 0.8778 | 0.9993 | 0.0007 | 0.1222 | 0.1985 | 0.9239 | 0.8960 |
| Camera Jitter | 0.7923 | 0.9924 | 0.0076 | 0.2077 | 1.5231 | 0.8520 | 0.8139 |
| Intermittent Object Motion | 0.6987 | 0.9946 | 0.0054 | 0.3013 | 3.2481 | 0.8146 | 0.7136 |
| Shadow | 0.9478 | 0.9914 | 0.0086 | 0.0521 | 1.0410 | 0.8585 | 0.8984 |
| Thermal | 0.7832 | 0.9945 | 0.0055 | 0.2168 | 1.2552 | 0.8922 | 0.8210 |

- [3] S. Y. Elhabian, K. M. El-Sayed, and S. H. Ahmed, "Moving object detection in spatial domain using background removal techniques-state-of-art," *Recent patents on computer science*, vol. 1, no. 1, pp. 32–54, 2008.
- [4] M. Cristani, M. Farenzena, D. Bloisi, and V. Murino, "Background subtraction for automated multisensor surveillance: a comprehensive review," *EURASIP Journal on Advances in signal Processing*, vol. 2010, p. 43, 2010.
- [5] T. Bouwmans, "Recent advanced statistical background modeling for foreground detection - a systematic survey," *Recent Patents on Computer Science*, vol. 4, no. 3, pp. 147–176, 2011.
- [6] A. Sobral and A. Vacavant, "A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos," *Computer Vision and Image Understanding*, vol. 122, no. 0, pp. 4–21, 2014.
- [7] Y. Benezeth, P.-M. Jodoin, B. Emile, C. Rosenberger, and H. Laurent, "Comparative study of background subtraction algorithms," *Journal of Electronic Imaging*, vol. 19, no. 3, pp. 033 003–033 003, 2010.
- [8] N. McFarlane and C. Schofield, "Segmentation and tracking of piglets in images," *Machine Vision and Applications*, vol. 8, no. 3, pp. 187–193, 1995.
- [9] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati, "Detecting moving objects, ghosts, and shadows in video streams," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 25, no. 10, pp. 1337–1342, 2003.
- [10] M.-H. Hung, J.-S. Pan, and C.-H. Hsieh, "Speed up temporal median filter for background subtraction," in *Pervasive Computing Signal Processing and Applications (PCSPA), 2010 First International Conference on*, 2010, pp. 297–300.
- [11] K. Karman and A. Brandt, "Moving object recognition using an adaptive background memory," *Time-Varying Image Processing and Moving Object Recognition*, vol. 2, pp. 297–307, 1990.
- [12] C. Ridder, O. Munkelt, and H. Kirchner, "Adaptive background estima-

TABLE VI
DETAILED EVALUATION RESULTS OF THE IUTIS-5 ALGORITHM FOR EACH CATEGORY OF THE EVALUATION DATASET.

| Scenarios | Recall | Specificity | FPR | FNR | PWC | Precision | FMeasure |
|----------------------------|--------|-------------|--------|--------|--------|-----------|----------|
| Overall | 0.7972 | 0.9952 | 0.0048 | 0.2028 | 1.0863 | 0.8105 | 0.7821 |
| Bad Weather | 0.7503 | 0.9994 | 0.0006 | 0.2497 | 0.4977 | 0.9349 | 0.8289 |
| Low Framerate | 0.8376 | 0.9974 | 0.0026 | 0.1624 | 0.7452 | 0.7724 | 0.7911 |
| Night Videos | 0.6333 | 0.9848 | 0.0152 | 0.3667 | 2.3252 | 0.4578 | 0.5132 |
| PTZ | 0.6687 | 0.9917 | 0.0083 | 0.3313 | 1.1465 | 0.4348 | 0.4703 |
| Turbulence | 0.7730 | 0.9999 | 0.0001 | 0.2270 | 0.1713 | 0.9624 | 0.8507 |
| Baseline | 0.9680 | 0.9983 | 0.0017 | 0.0320 | 0.3053 | 0.9464 | 0.9567 |
| Dynamic Background | 0.8636 | 0.9996 | 0.0004 | 0.1364 | 0.1808 | 0.9324 | 0.8902 |
| Camera Jitter | 0.8220 | 0.9925 | 0.0075 | 0.1780 | 1.4389 | 0.8511 | 0.8332 |
| Intermittent Object Motion | 0.7047 | 0.9963 | 0.0037 | 0.2953 | 3.0420 | 0.8501 | 0.7296 |
| Shadow | 0.9492 | 0.9923 | 0.0077 | 0.0508 | 0.9484 | 0.8766 | 0.9084 |
| Thermal | 0.7990 | 0.9952 | 0.0048 | 0.2010 | 1.1484 | 0.8969 | 0.8303 |

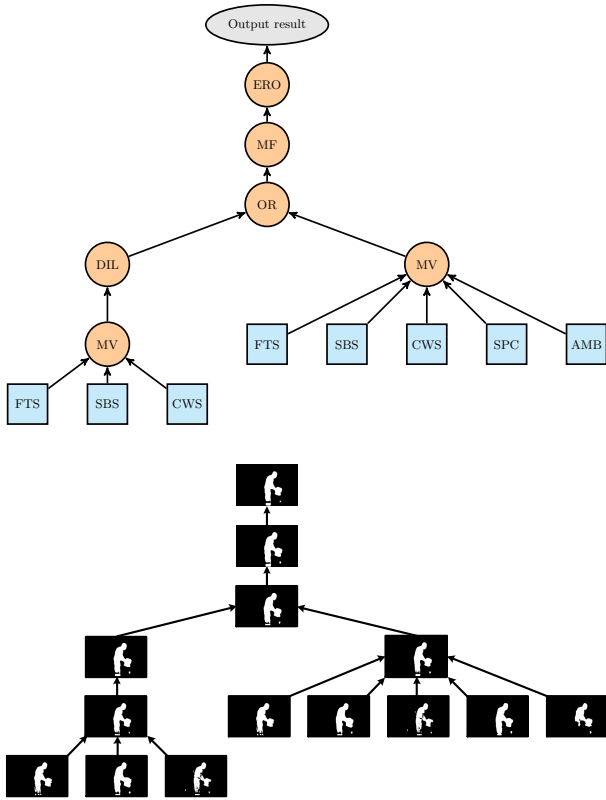


Fig. 3. IUTIS-5 solution tree and its example masks. SBS, FTS, CWS, SPC and AMB refer to SuBSENSE, FTSG, CwsarDH, Spectral-360 and AMBER algorithm respectively.

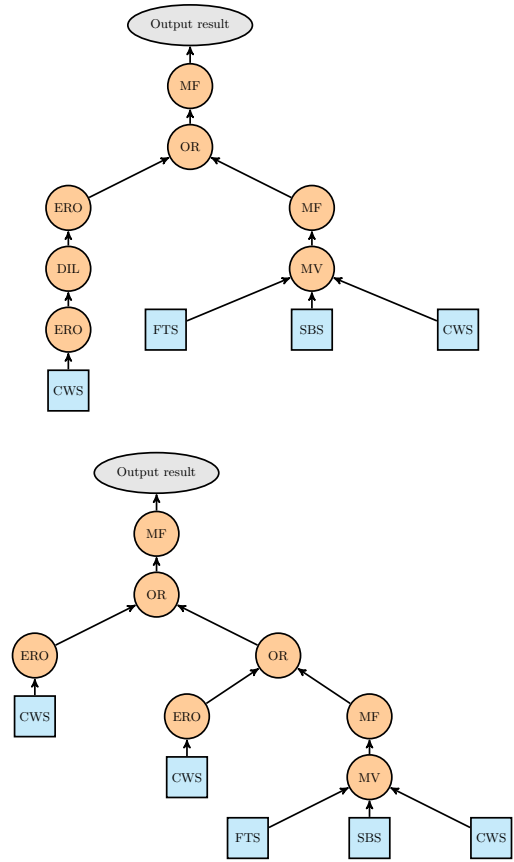


Fig. 4. Two variant solutions of IUTIS-3 found by GP. These were discarded having size greater than IUTIS-3. SBS, FTS, and CWS refer to SuBSENSE, FTSG, and CwsarDH algorithm respectively.

tion and foreground detection using kalman-filtering,” in *Proceedings of International Conference on recent Advances in Mechatronics*, 1995, pp. 193–199.

- [13] S. C. Sen-Ching and C. Kamath, “Robust techniques for background subtraction in urban traffic video,” in *Electronic Imaging 2004*. International Society for Optics and Photonics, 2004, pp. 881–892.
- [14] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, “Wallflower: principles and practice of background maintenance,” in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, vol. 1, 1999, pp. 255–261.
- [15] J. Zheng, Y. Wang, N. Nihan, and M. Hallenbeck, “Extracting roadway background image: Mode-based approach,” *Transportation Research Record: Journal of the Transportation Research Board*, no. 1944, pp. 82–88, 2006.

- [16] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, “Pfinder: Real-time tracking of the human body,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 19, no. 7, pp. 780–785, 1997.
- [17] C. Stauffer and W. Grimson, “Adaptive background mixture models for real-time tracking,” in *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on*, vol. 2, 1999, pp. 246–252.
- [18] H. Kim, R. Sakamoto, I. Kitahara, T. Toriyama, and K. Kogure, “Robust foreground extraction technique using background subtraction with multiple thresholds,” *Optical Engineering*, vol. 46, no. 9, pp. 097 004–097 004, 2007.
- [19] M. S. Allili, N. Bouguila, and D. Ziou, “Finite general gaussian mixture modeling and application to image and video foreground segmentation,”



Fig. 5. Examples of binary masks created by the tested algorithms. The superscripts indicate in what fusion set \mathcal{C} the algorithm is used (e.g. SuBSENSE, FTSG, and CwisarDH are used to build IUTIS-3, MV-3, PROF-3 and STAPLE-3).

TABLE VII
AVERAGE RANKING OF THE ALGORITHMS OF THE
CHANGEDetection.NET WEBSITE AS PER NOV 5 2015.

| Method | Average ranking across categories |
|--------------------------------------------|--------------------------------------|
| IUTIS-5 | 2.27 |
| IUTIS-3 | 5.00 |
| PAWCS [71] | 5.91 |
| SuBSENSE [31] | 6.27 |
| FTSG [24] | 7.55 |
| SaliencySubsense [*] | 8.36 |
| Superpixel Strengthen Backgr. Subtr. [*] | 8.36 |
| M4CD Version 1.5 [*] | 9.82 |
| Multimode Backgr. Subtr. [*] | 10.73 |
| M4CD Version 1.0 [*] | 10.82 |
| C-EFIC [*] | 10.91 |
| Multimode Backgr. Subtr. V.0 (MBS V0) [72] | 12.64 |
| EFIC [73] | 13.00 |
| CwisarDH [67] | 13.45 |
| Spectral-360 [43] | 15.55 |
| AMBER [32] | 17.27 |
| AAPSA [74] | 18.36 |
| GraphCutDiff [75] | 20.82 |
| KNN [26] | 20.82 |
| SC_SOBS [41] | 20.82 |
| RMoG [68] | 21.18 |
| Mahalanobis distance [7] | 21.45 |
| SOBS_CF [76] | 21.64 |
| KDE [69] | 22.91 |
| CP3-online [77] | 24.55 |
| GMM Stauffer & Grimson [17] | 24.91 |
| GMM Zivkovic [78] | 25.73 |
| Multiscale Spatio-Temporal BG Model [79] | 27.55 |
| Euclidean distance [7] | 29.09 |

Note: Methods with reference[*] have been submitted to journals or conferences. See the changedetection.net website for current status.

- Journal of Electronic Imaging*, vol. 17, no. 1, pp. 013 005–013 005, 2008.
- [20] Y. Sheikh and M. Shah, “Bayesian modeling of dynamic scenes for object detection,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 11, pp. 1778–1792, 2005.
- [21] O. Tuzel, F. Porikli, and P. Meer, “A bayesian approach to background modeling,” in *Computer Vision and Pattern Recognition-Workshops, 2005. CVPR Workshops. IEEE Computer Society Conference on. IEEE*, 2005, pp. 58–58.
- [22] L. Li, W. Huang, I.-H. Gu, and Q. Tian, “Statistical modeling of complex backgrounds for foreground object detection,” *Image Processing, IEEE Transactions on*, vol. 13, no. 11, pp. 1459–1472, 2004.
- [23] C. Benedek and T. Szirányi, “Bayesian foreground and shadow detection in uncertain frame rate surveillance videos,” *Image Processing, IEEE Transactions on*, vol. 17, no. 4, pp. 608–621, 2008.
- [24] R. Wang, F. Bunyak, G. Seetharaman, and K. Palaniappan, “Static and moving object detection using flux tensor with split gaussian models,” in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on. IEEE*, 2014, pp. 420–424.
- [25] A. Elgammal, R. Duraiswami, D. Harwood, and L. S. Davis, “Background and foreground modeling using nonparametric kernel density estimation for visual surveillance,” *Proceedings of the IEEE*, vol. 90, no. 7, pp. 1151–1163, 2002.
- [26] Z. Zivkovic and F. van der Heijden, “Efficient adaptive density estimation per image pixel for the task of background subtraction,” *Pattern Recognition Letters*, vol. 27, no. 7, pp. 773–780, 2006.
- [27] T. Tanaka, A. Shimada, D. Arita, and R.-i. Taniguchi, “A fast algorithm for adaptive background model construction using parzen density estimation,” in *Advanced Video and Signal Based Surveillance, 2007. AVSS 2007. IEEE Conference on. IEEE*, 2007, pp. 528–533.
- [28] Y. Nonaka, A. Shimada, H. Nagahara, and R.-i. Taniguchi, “Evaluation report of integrated background modeling based on spatio-temporal features,” in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on. IEEE*, 2012, pp. 9–14.
- [29] O. Barnich and M. Van Droogenbroeck, “Vibe: A universal background subtraction algorithm for video sequences,” *Image Processing, IEEE Transactions on*, vol. 20, no. 6, pp. 1709–1724, 2011.
- [30] P.-L. St-Charles, G.-A. Bilodeau, and R. Bergevin, “Flexible background subtraction with self-balanced local sensitivity,” pp. 414–419, 2014.
- [31] —, “Subsense: A universal change detection method with local adaptive sensitivity,” *Image Processing, IEEE Transactions on*, vol. 24, no. 1, pp. 359–373, 2015.
- [32] B. Wang and P. Dudek, “A fast self-tuning background subtraction algorithm,” in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on. IEEE*, 2014, pp. 401–404.
- [33] N. Oliver, B. Rosario, and A. Pentland, “A bayesian computer vision system for modeling human interactions,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 8, pp. 831–843, 2000.
- [34] Y. Li, “On incremental and robust subspace learning,” *Pattern Recognition*, vol. 37, no. 7, pp. 1509–1518, 2004.
- [35] S. S. Bucak and B. Günsel, “Incremental subspace learning via non-negative matrix factorization,” *Pattern Recognition*, vol. 42, no. 5, pp. 788–797, 2009.
- [36] N. Wang, T. Yao, J. Wang, and D.-Y. Yeung, “A probabilistic approach to robust matrix factorization,” in *Computer Vision ? ECCV 2012*, ser. Lecture Notes in Computer Science, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, Eds. Springer Berlin Heidelberg, 2012, vol. 7578, pp. 126–139.
- [37] H.-H. Lin, T.-L. Liu, and J.-H. Chuang, “Learning a scene background model via classification,” *Signal Processing, IEEE Transactions on*, vol. 57, no. 5, pp. 1641–1654, 2009.
- [38] B. Han and L. Davis, “Density-based multifeature background subtraction with support vector machine,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 5, pp. 1017–1023, 2012.
- [39] L. Cheng, M. Gong, D. Schuurmans, and T. Caelli, “Real-time discriminative background subtraction,” *Image Processing, IEEE Transactions on*, vol. 20, no. 5, pp. 1401–1414, 2011.
- [40] D. Culibrk, O. Marques, D. Socek, H. Kalva, and B. Furht, “Neural network approach to background modeling for video object segmentation,” *Neural Networks, IEEE Transactions on*, vol. 18, no. 6, pp. 1614–1627, 2007.
- [41] L. Maddalena and A. Petrosino, “The sob's algorithm: what are the limits?” in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on. IEEE*, 2012, pp. 21–26.
- [42] A. Ferone and L. Maddalena, “Neural background subtraction for pan-tilt-zoom cameras,” *Systems, Man, and Cybernetics: Systems, IEEE Transactions on*, vol. 44, no. 5, pp. 571–579, 2014.
- [43] M. Sedky, M. Moniri, and C. C. Chibelushi, “Spectral-360: A physics-based technique for change detection,” in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on. IEEE*, 2014, pp. 405–408.
- [44] D. H. Parks and S. S. Fels, “Evaluation of background subtraction algorithms with post-processing,” in *Advanced Video and Signal Based Surveillance, 2008. AVSS'08. IEEE Fifth International Conference on. IEEE*, 2008, pp. 192–199.
- [45] V. Pham, P. Vo, V. T. Hung, and L. H. Bac, “Gpu implementation of extended gaussian mixture model for background subtraction,” in *Computing and Communication Technologies, Research, Innovation, and Vision for the Future (RIVF), 2010 IEEE RIVF International Conference on. Nov 2010*, pp. 1–4.
- [46] L. Cheng, M. Gong, D. Schuurmans, and T. Caelli, “Real-time discriminative background subtraction,” *Image Processing, IEEE Transactions on*, vol. 20, no. 5, pp. 1401–1414, 2011.
- [47] G. Szwoch, D. Ellwart, and A. Czyewski, “Parallel implementation of background subtraction algorithms for real-time video processing on a supercomputer platform,” *Journal of Real-Time Image Processing*, pp. 1–15, 2012.
- [48] Y. Wang, P.-M. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, and P. Ishwar, “Cdnets 2014: An expanded change detection benchmark dataset,” in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on. IEEE*, 2014, pp. 393–400.
- [49] N. Goyette, P.-M. Jodoin, F. Porikli, J. Konrad, and P. Ishwar, “A novel video dataset for change detection benchmarking,” *Image Processing, IEEE Transactions on*, vol. 23, no. 11, pp. 4663–4679, 2014.
- [50] N. Goyette, P. Jodoin, F. Porikli, J. Konrad, and P. Ishwar, “Changetection.net: A new change detection benchmark dataset,” in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, 2012, pp. 1–8.
- [51] P.-M. Jodoin, S. Pierard, Y. Wang, and M. Van Droogenbroeck, “Overview and benchmarking of motion detection methods,” *Back-*

- ground Modeling and Foreground Detection for Video Surveillance*, 2014.
- [52] J. Kittler, M. Hatef, R. Duin, and J. Matas, "On combining classifiers," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 20, no. 3, pp. 226–239, 1998.
 - [53] L. I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*. Wiley-Interscience, 2004.
 - [54] D. Ruta and B. Gabrys, "An overview of classifier fusion methods," *Computing and Information systems*, vol. 7, no. 1, pp. 1–10, 2000.
 - [55] —, "Classifier selection for majority voting," *Information Fusion*, vol. 6, no. 1, pp. 63–81, 2005, diversity in Multiple Classifier Systems.
 - [56] H. Wang, Y. Zhang, and G. Qian, "Multiple binary classifiers fusion using induced intuitionistic fuzzy ordered weighted average operator," in *Information and Automation (ICIA), 2011 IEEE International Conference on*, 2011, pp. 230–235.
 - [57] N. García-Pedrajas and D. Ortiz-Boyer, "An empirical study of binary classifier fusion methods for multiclass classification," *Information Fusion*, vol. 12, no. 2, pp. 111–130, 2011.
 - [58] S. Aljahdali and E. Zanaty, "Combining multiple segmentation methods for improving the segmentation accuracy," in *Computers and Communications, 2008. ISCC 2008. IEEE Symposium on*, 2008, pp. 649–653.
 - [59] D. Ruta and B. Gabrys, "An overview of classifier fusion methods," *Computing and Information systems*, vol. 7, no. 1, pp. 1–10, 2000.
 - [60] S. K. Warfield, K. H. Zou, and W. M. Wells, "Validation of image segmentation and expert quality with an expectation-maximization algorithm," in *Medical Image Computing and Computer-Assisted Intervention?MICCAI 2002*. Springer, 2002, pp. 298–306.
 - [61] —, "Simultaneous truth and performance level estimation (staple): an algorithm for the validation of image segmentation," *Medical Imaging, IEEE Transactions on*, vol. 23, no. 7, pp. 903–921, 2004.
 - [62] T. Rohlfing, D. B. Russakoff, and C. R. Maurer Jr, "Performance-based classifier combination in atlas-based image segmentation using expectation-maximization parameter estimation," *Medical Imaging, IEEE Transactions on*, vol. 23, no. 8, pp. 983–994, 2004.
 - [63] M. Mignotte, "A label field fusion bayesian model and its penalized maximum rand estimator for image segmentation," *Image Processing, IEEE Transactions on*, vol. 19, no. 6, pp. 1610–1624, 2010.
 - [64] —, "Segmentation by fusion of histogram-based-means clusters in different color spaces," *Image Processing, IEEE Transactions on*, vol. 17, no. 5, pp. 780–787, 2008.
 - [65] H. Wang, Y. Zhang, R. Nie, Y. Yang, B. Peng, and T. Li, "Bayesian image segmentation fusion," *Knowledge-Based Systems*, vol. 71, pp. 162–168, 2014.
 - [66] J. R. Koza, *Genetic programming: on the programming of computers by means of natural selection*. MIT press, 1992, vol. 1.
 - [67] M. D. Gregorio and M. Giordano, "Change detection with weightless neural networks," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on*. IEEE, 2014, pp. 409–413.
 - [68] S. Varadarajan, P. Miller, and H. Zhou, "Spatial mixture of gaussians for dynamic background modelling," in *Advanced Video and Signal Based Surveillance (AVSS), 2013 10th IEEE International Conference on*. IEEE, 2013, pp. 63–68.
 - [69] A. Elgammal, D. Harwood, and L. Davis, "Non-parametric model for background subtraction," in *Computer Vision/ECCV 2000*. Springer, 2000, pp. 751–767.
 - [70] L. Davis, "Adapting operator probabilities in genetic algorithms," in *International Conference on Genetic Algorithms'89*, 1989, pp. 61–69.
 - [71] P.-L. St-Charles, G.-A. Bilodeau, and R. Bergevin, "A self-adjusting approach to change detection based on background word consensus," in *Applications of Computer Vision (WACV), 2015 IEEE Winter Conference on*. IEEE, 2015, pp. 990–997.
 - [72] H. Sajid and S.-C. S. Cheung, "Background subtraction for static & moving camera," in *IEEE International Conference on Image Processing*, 2015.
 - [73] G. Allebosch, F. Deboeverie, P. Veelaert, and W. Philips, "Efic: Edge based foreground background segmentation and interior classification for dynamic camera viewpoints," in *Advanced Concepts for Intelligent Vision Systems*, ser. Lecture Notes in Computer Science, S. Battiato, J. Blanc-Talon, G. Gallo, W. Philips, D. Popescu, and P. Scheunders, Eds. Springer International Publishing, 2015, vol. 9386, pp. 130–141.
 - [74] G. Ramirez-Alonso and M. I. Chacn-Murgua, "Auto-adaptive parallel {SOM} architecture with a modular analysis for dynamic object segmentation in videos," *Neurocomputing*, p. Accepted, 2015.
 - [75] A. Miron and A. Badii, "Change detection based on graph cuts," in *IWSSIP 2015*, 2015.
 - [76] L. Maddalena and A. Petrosino, "A fuzzy spatial coherence-based approach to background/foreground separation for moving object detection," *Neural Computing and Applications*, vol. 19, no. 2, pp. 179–186, 2010.
 - [77] D. Liang and S. Kaneko, "Improvements and experiments of a compact statistical background model," *arXiv preprint arXiv:1405.6275*, 2014.
 - [78] Z. Zivkovic, "Improved adaptive gaussian mixture model for background subtraction," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, vol. 2, Aug 2004, pp. 28–31.
 - [79] X. Lu, "A multiscale spatio-temporal background model for motion detection," in *Image Processing (ICIP), IEEE International Conference on*. IEEE, 2014.